



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/706,177	11/12/2003	Akram A. Bou-Ghannam	BOC9-2003-0046 (417)	6117
40987	7590	08/22/2007		
AKERMAN SENTERFITT P. O. BOX 3188 WEST PALM BEACH, FL 33402-3188				
			EXAMINER HOLLOWAY, DAVID A	
			ART UNIT 2195	PAPER NUMBER
			MAIL DATE 08/22/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/706,177

Applicant(s)

BOU-GHANNAM ET AL.

Examiner

David A. Holloway

Art Unit

2109

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 12 November 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-38 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-38 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 12 November 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☒ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 20040326.
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- ☐ Notice of Informal Patent Application
- ☐ Other: _____.

DETAILED ACTION

1. Claims 1-38 are pending in this application.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to

Art Unit: 2109

be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 14-15 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 11-13 of copending Application No. 10/705,990. Although the conflicting claims are not identical, they are not patentably distinct from each other.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

The subject matter claimed in the instant application is fully disclosed in the referenced copending application and would be covered by any patent granted on the copending application since the referenced copending application and the instant application are claiming common subject matter, as follows in Table1 below.

Instant application	Limitations	10/705,990	Limitations
Claim 15	A system for processing complex requests for Web services comprising:	Claim 13	A system for processing complex requests for Web services comprising:

Art Unit: 2109

	A server configured to receive a request for more than one Web service,		A hypertext transfer protocol server, configured to receive a single request for more than one Web service,
	At least one servlet configured to extract a pattern object from the request and to format a response to the request,		At least one servlet configured to extract a pattern object from the request and to format a response to the request,
	A common memory that temporarily stores the pattern object while the Web services specified by the pattern object execute, wherein said common memory generates events when content of said common memory is changed,		A common memory that temporarily stores the pattern object while the Web services specified by the pattern object execute
	A plurality of watchers, each watcher corresponding to a particular Web service, and		A plurality of watchers, each watcher being associated with a particular Web service
	A termination watcher configured to provide the pattern object back to one of said plurality of servlets to generate a response		A termination watcher configured to detect termination criterion, wherein the termination watcher provides the pattern object back to one of said plurality of servlets to generate a response upon detecting the termination criterion
	A scheduler configured to receive events, compare trigger conditions specified by said service activation rules with trigger information of events, compare state conditions of said service activation rules in the trigger list with a state of the common memory, and select at least one of said service activation rules causing the invocation of at least one of said watchers and corresponding Web services		Cache logic for comparing pattern objects from received requests with cached pattern objects, wherein said cache logic causes each watcher in a list of watchers associated with a cached pattern object that matches the pattern object from a received request to invoke an associated Web service

Table 1

Art Unit: 2109

4. It would have been obvious to one skilled in the art at the time of invention to modify claim 11 of the instant application to add service activation rules in order to be able to determine which Web service to invoke. The dependent claim 15 of the instant application and claim 13 of the copending Application No. 10/705,990 are rejected in the analysis above.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 13, 25, and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shim in view of Xie.

7. As to claim 13, Shim discloses a method of resolving conflicts between competing Web services comprising:
reading an execution list of service activation rules corresponding to watchers (page 1, [0008], lines 5-7, the service activation rules are read from the execution list to invoke the watchers, ultimately to invoke the Web services),
identifying service activation rules in the execution list corresponding to competitive Web services (page 1, [0008], lines 5-7, the agent module receives and analyzes user commands, so the agent module is able to identify the server activation rules),

invoking watchers specified by the identified service activation rules according to said comparing step (page 1, [0008], lines 5-7, the agent module also performs operations according the results of the analysis of the commands, implying that tasks associated with the matching rules are invoked and that these tasks invoke the appropriate Web service. The tasks are the watchers).

8. Shim does not disclose that the service activation rules in the execution list correspond to competitive Web services, nor does Shim disclose comparing the identified service activation rules with at least one service selection rule. However, Xie discloses that the service activation rules in the execution list correspond to competitive Web services (page 6, [0070], lines 10-13). Xie also discloses comparing the identified service activation rules with at least one service selection rule (page 6, [0070], lines 10-13).

9. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shin and Xie before him or her to incorporate the service selection rule feature of Xie in the system of Shim, allowing the user to specify preferred Web services to employ.

10. As to claim 25, Shim discloses a system for resolving conflicts between competing Web services comprising:

means for reading an execution list of service activation rules corresponding to watchers, wherein each watcher is configured to invoke an associated Web

Art Unit: 2109

service (page 1, [0008], lines 5-7, the agent module contains the watchers), means for identifying service activation rules in the execution list corresponding to competitive Web services (page 1, [0008], lines 5-7, the agent module receives and analyzes user commands, so the agent module is able to identify the server activation rules), and means for invoking watchers specified by the identified service activation rules according to said means for comparing (page 1, [0008], lines 5-7, the agent module also performs operations according the results of the analysis of the commands, implying that tasks associated with the matching rules are invoked and that these tasks invoke the appropriate Web service. The tasks are the watchers).

11. Shim does not disclose means for comparing the identified service activation rules with at least one service selection rule. However, Xie discloses means for comparing the identified service activation rules with at least one service selection rule (page 6, [0070], lines 10-13).

12. Shim and Xie are analogous art because they are in the same field of endeavor of Web-based service systems.

13. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shin and Xie before him or her to incorporate the service selection rule feature of Xie in the system of Shim, allowing the user to specify preferred Web services to employ.

14. As to claim 38, Shim discloses a machine readable storage, having stored

Art Unit: 2109

thereon a computer program having a plurality of code sections executable by a machine for causing the machine to perform the steps of:

reading an execution list of service activation rules corresponding to watchers (page 1, [0008], lines 5-7, the service activation rules are read from the execution list to invoke the watches, ultimately to invoke the Service), wherein each watcher is configured to invoke an associated Web service (page 1, [0008], lines 5-7, the agent module contains the watchers),

identifying service activation rules in the execution list corresponding to competitive Web services (page 1, [0008], lines 5-7, the agent module receives and analyzes user commands, so the agent module is able to identify the server activation rules), invoking watchers specified by the identified service activation rules according to said comparing step (page 1, [0008], lines 5-7, the agent module also performs operations according the results of the analysis of the commands, implying that tasks associated with the matching rules are invoked and that these tasks invoke the appropriate Web service. The tasks are the watchers).

15. Shim does not disclose comparing the identified service activation rules with at least one service selection rule. However, Xie discloses comparing the identified service activation rules with at least one service selection rule (page 6, [0070], lines 10-13).

16. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shin and Xie before him or her to incorporate the

Art Unit: 2109

service selection rule feature of Xie in the system of Shim, allowing the user to specify preferred Web services to employ.

17. Claims 1-11, 14-16, 19, 20, 23, and 26-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shim (US 2001/0020249), in view of Auer et al. (US 6,067,637), hereinafter Auer, and further in view of Wu (US 2005/0038708).

18. As to claim 1, Shim discloses a method of scheduling a request for a plurality of Web services comprising providing a plurality of service activation rules for causing a watcher to invoke a particular Web service (Abstract, lines 1-8, the Web services are scheduled based on the analysis of the user commands, which implies service activation rules and a watcher that is able to determine when to invoke a particular Web service).

19. However, Shim does not disclose "each service activation rule specifying trigger and state conditions", nor does Shim disclose steps (b) through (f). On the other hand, Auer discloses that each server activation rule specifies trigger and state conditions (col. 6, lines 49-51).

20. Auer discloses that the method comprises:

(b) receiving at least one event indicating a change in a common memory, wherein each event specifies trigger information (Abstract, lines 8-11, the event trigger filters data when data is sent over the communications link. Col. 5, lines 66-67, col. 6, lines 1-4, data is sent from the remote system when modifications are made to the remote screen,

Art Unit: 2109

i.e., remote screen memory (Although the remote screen memory is not a common memory, whether or not the screen memory is in a common memory or not is not critical to the operability of the system of Auer, see claim 8). The event specifies trigger information, since a processor compares the match portion of at least one of the event triggers, see Abstract, lines 11-12), and

(c) comparing the trigger conditions of the service activation rules

with the trigger information of the at least one event (Abstract, lines 11-12),

(d) adding service activation rules that match the at least one

event to a trigger list (col. 2, lines 27-30),

(e) comparing the state conditions of service activation rules in the trigger list with a state of the common memory (col. 6, lines 38-41, the "pattern portion" corresponds to the state condition of the rule, col. 6 lines 49-51), and

(f) selecting the service activation rules of the trigger list that match the state of the common memory (col. 6, lines 51-52, the action portion of the rule is executed).

21. Shim and Auer are analogous art because they are in the same field of endeavor of computer systems.

22. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim and Auer before him or her to incorporate the form of the rules used by Auer in the system of Shim, to incorporate the indication of an event when a change in a common memory occurs, and to incorporate the comparison of the trigger conditions of the service activation rules with the trigger information of the

event(s), to incorporate the comparison of state conditions of the service activation rules in the trigger list with a state of the common memory, and to incorporate the selection of service activation rules of the trigger list that match the state of the common memory.

The motivation for doing so would have been to enable the rules to specify the conditions that the rules-based system would respond to based on the contents of a working memory as taught by Auer (col. 6, lines 49-51), and to allow the system to execute the action portion of the rule as taught by Auer (Abstract, lines 11-14), and to allow a second comparison to be used in the rule matching process as taught by Auer (col. 6, lines 49-54).

23. Neither Shim nor Auer disclose that selecting the service activation rules of the trigger list that match the state of the common memory causes the invocation of at least one watcher and corresponding Web service. However, in the combined system of Shim and Auer the invocation of a watcher and corresponding Web service will result from the agent module performing commands based on analyzing the user commands (Shim, page 1, [0008], lines 5-7).

24. Neither Shim nor Auer discloses that the state of the common memory is dictated by at least one pattern object. However, Wu discloses that the state of the common memory is dictated by at least one pattern object (p. 4, [0046], for each user submission concurrency configurations are saved, so data is written to memory in the form of a pattern object, also see p. 1, [0019], first sentence)

Art Unit: 2109

25. Shim, Auer, and Wu are analogous arts because they are in the same field of endeavor of computer systems.

26. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim and Auer, and the teachings of Wu before him or her to write data to memory in the form of a pattern object, i.e., have at least one pattern object dictate the state of the common memory. The motivation for doing so would have been to keep the data in memory organized which, in turn, makes it easier to work with the data.

27. As to claim 26, Shim discloses a machine readable storage, having stored thereon a computer program having a plurality of code sections executable by a machine for causing the machine to perform the steps of:

(a) providing a plurality of service activation rules, each service activation rule specifying a trigger condition and a state condition for causing a watcher to invoke a particular Web service (Abstract, lines 1-8).

28. Shim does not disclose that that the server activation rules specify trigger and state conditions nor does Shim discloses steps (b) through (f). However, Auer discloses that the server activation rules specify trigger and state conditions (col. 6, lines 49-51).

29. Auer also discloses:

(b) receiving at least one event indicating a change in a common memory, wherein each event specifies trigger information (Abstract, lines 8-11, the event trigger filters data

Art Unit: 2109

when data is sent over the communications link. Col. 5, lines 66-67, col. 6, lines 1-4, data is sent from the remote system when modifications are made to the remote screen, i.e., remote screen memory (Although the remote screen memory is not a common memory, whether or not the screen memory is in a common memory or not is not critical to the operability of the system of Auer, see claim 8). The event specifies trigger information, since a processor compares the match portion of at least one of the event triggers, see Abstract, lines 11-12).

(c) comparing the trigger conditions of the service activation rules with the trigger information of the at least one event (Abstract, lines 11-12),

(d) adding service activation rules that match the at least one event to a trigger list (col. 2, lines 27-30, adding the rules to a list is just an intermediate),

(e) comparing the state conditions of service activation rules in the trigger list with a state of the common memory (col. 6, lines 38-41, the "pattern portion" corresponds to the state condition of the rule, col. 6 lines 49-51), and

(f) selecting the service activation rules of the trigger list that match the state of the common memory, thereby causing the invocation of at least one watcher and corresponding Web service (col. 6, lines 51-52, the action portion of the rule is executed)

30. It would have been obvious to a person having ordinary skill in the art at the time

Art Unit: 2109

of invention having the teachings of Shim and Auer before him or her to incorporate the form of the rules used by Auer in the system of Shim, to incorporate the indication of an event when a change in a common memory occurs, and to incorporate the comparison of the trigger conditions of the service activation rules with the trigger information of the event(s), to incorporate the comparison of state conditions of the service activation rules in the trigger list with a state of the common memory, and to incorporate the selection of service activation rules of the trigger list that match the state of the common memory.

The motivation for doing so would have been to enable the rules to specify the conditions that the rules-based system would respond to based on the contents of a working memory as taught by Auer (col. 6, lines 49-51), and to allow the system to execute the action portion of the rule as taught by Auer (Abstract, lines 11-14), and to allow a second comparison to be used in the rule matching process as taught by Auer (col. 6, lines 49-54).

31. Neither Shim nor Auer discloses that the state of the common memory is dictated by at least one pattern object. However, Wu discloses that the state of the common memory is dictated by at least one pattern object (p. 4, [0046], for each user submission concurrency configurations are saved, so data is written to memory in the form of a pattern object, also see p. 1, [0019], first sentence)

32. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim and Auer, and the teachings of Wu before him or her to write data to memory in the form of a pattern object, i.e., have at least one

Art Unit: 2109

pattern object dictate the state of the common memory. The motivation for doing so would have been to keep the data in memory organized which, in turn, makes it easier to work with the data. For example, if other servlets need to access the data, they can deserialize the object to easily access the object members.

33. As to claims 2 and 27, neither Shim, Auer, nor Wu specifically disclose the method of claim 1, wherein each event indicating a change in the common memory is associated with the pattern object. However, it would have been obvious to one skilled in the art at the time of invention that an event indicating a change in the common memory is associated with the pattern object, since the data in memory is written in the form of pattern objects.

34. As to claims 3 and 28, neither Shim, Auer, nor Wu specifically disclose the method of claim 1, said step (e) further comprising the step of adding service activation rules of the trigger list that match the pattern object to an executable list, wherein each service activation rule in the executable list is executed in said step (f). However, It would have been obvious to one skilled in the art at the time of invention to put the service activation rules of the tasks to be executed in an executable list in order to track the Web services that are being invoked.

35. As to claims 4 and 29, the claims are rejected for the same reasons as claims 1 and 26 above. In addition, Auer discloses the method of claim 1, further comprising the steps of:

at least one of the watchers modifying the common memory (col. 1, lines 51-55, the

Art Unit: 2109

event triggers update the working memory), the common memory sending at least one event indicating a state change (col. 5, lines 66-67, col. 6, lines 1-4, data is sent over the communications link specifying that data in the screen memory has changed), wherein each event specifies a trigger condition (col. 1, lines 48-56, the event triggers examine the data sent over the communications link to see if it matches the trigger condition of one of the rules, so the information sent over the communications link comprises trigger conditions), and repeating said steps (b)-(f) (col. 1, lines 39-43, the steps are repeated each time data is sent over the communications link).

36. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim and Auer before him or her for the method to further comprise at least one of the watchers modifying the common memory, for the method to have the common memory send at least one event indicating a state change, for the method to further comprise for each event specifying a trigger condition in order for the expert system to use the trigger condition in the rule matching process, and for the method to repeat said steps (b) through (f) in order to continue tracking the status of the Web services that have run and to continue executing the automatic operations resulting from the user commands. The motivation for doing so would have been to have the memory reflect the status of the system relative to the action invoked by the watcher, to allow the expert system to check the status returned by the watcher (which returns status from the Web service) to decide what action to take next, to enable the expert system to use the trigger condition in the rule matching process, and to continue tracking the status of the Web services that have run and allow the system to continue

executing the automatic operations resulting from the user commands (Shim, col. 1, lines 7-9).

37. As to claims 5 and 30, neither Shim nor Auer discloses the method of claim 4, wherein the at least one of the watchers modifies the common memory according to instructions from an associated one of the Web services. However, Wu discloses the method of claim 4, wherein the at least one of the watchers modifies the common memory according to instructions from an associated one of the Web services (page 2, [0024], the invention dynamically reconfigures Web services by intercepting, transforming, and redirecting SOAP messages, and the SOAP messages are in memory).

38. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer, and Wu before him or her to have at least one of the watchers modify the common memory according to instructions from an associated one of the Web services in order to report status back from the Web service.

39. As to claims 6 and 31, neither Shim nor Auer discloses the method of claim 5, wherein at least one of the watchers modifies the common memory by modifying the pattern object. However, Wu discloses the method of claim 5, wherein at least one of the watchers modifies the common memory by modifying the pattern object (page 2, [0024], the invention dynamically reconfigures Web services by intercepting, transforming, and redirecting SOAP messages, and the SOAP messages are in memory).

Art Unit: 2109

40. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer and Wu before him or her to have one of the watchers modify the common memory by modifying the pattern object. The motivation for doing so would have been to keep the changes encapsulated in an object, making the data easier to manage.

41. As to claims 7 and 32, neither Shim nor Auer discloses the method of claim 1, wherein each pattern object specifies at least two Web services to be performed. However, Wu discloses the method of claim 1, wherein each pattern object specifies at least two Web services to be performed (page 2, [0023], when the invention is configured as a Web services tester, the number of invocations per thread is set to be more than one. This is interpreted as meaning that there are more than one Web services to be performed for the given thread and that the corresponding pattern object will specify at least two Web services).

42. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer, and Wu before him or her to reduce the number of pattern objects needed to describe multiple Web service requests in order to improve performance of the system.

43. As to claims 8 and 33, neither Shim nor Auer discloses the method of claim 1, wherein at least two watchers each invoke an associated Web service to operate concurrently with one another in said step (f). However, Wu discloses the method of claim 1, wherein at least two watchers each invoke an associated Web service to operate concurrently with one another in said step (f) (page 2, [0037], lines 1-7,

Art Unit: 2109

concurrent Web Services invocation requests are described). It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer, and Wu before him or her to allow at least two watchers to invoke an associated Web service to operate concurrently in order to improve system performance.

44. As to claims 9 and 34, neither Shim nor Auer disclose the method of claim 1, wherein at least two watchers each invoke an associated Web service to operate sequentially in said step (f). However, Wu discloses the method of claim 1, wherein at least two watchers each invoke an associated Web service to operate sequentially in said step (f) (page 3, [0082], invoking one of the Web services is described, especially, "the sequence in the invocation list dictate what operations are to be invoked and what the invocation sequence is"). It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer and Wu before him or her to allow at least two watchers to invoke an associated Web service to operate sequentially in order to improve system performance.

45. As to claims 10 and 35, neither Shim nor Auer disclose the method of claim 1, further comprising watchers continuing to invoke Web services until a termination watcher is activated and removes the pattern object from the common memory. However, Wu discloses the method of claim 1, further comprising watchers continuing to invoke Web services until a termination watcher is activated and removes the pattern object from the common memory (page 3, [0046], the queue watcher removes the request from the queue, also see Figure 6).

46. It would have been obvious to a person having ordinary skill in the art at the time

Art Unit: 2109

of invention having the teachings of Shim, Auer, and Wu before him or her to have the watchers continue to invoke Web services until a termination watcher is activated and removes the pattern object from the common memory. The motivation for doing so would have been to allow users to continue to submit requests and have their requests carried out until such time where there are no more requests are being accepted.

47. As to claims 11 and 36, neither Shim nor Auer discloses the method of claim 1, further comprising the step of at least one of the watchers modifying the pattern object according to instructions from an associated one of the Web services. However, Wu discloses the method of claim 1, further comprising the step of at least one of the watchers modifying the pattern object according to instructions from an associated one of the Web services (page 2, [0024], the invention dynamically reconfigures Web services by intercepting, transforming, and redirecting SOAP messages).

48. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer, and Wu before him or her to have at least one of the watchers modify the pattern object according to instructions from an associated one of the Web services. The motivation for doing so would have been to return data back from the Web service, and the pattern object is associated with Web services.

49. As to claim 14, Shim discloses a system for processing complex requests for Web services comprising:

a plurality of service activation rules (Abstract, lines 1-8, service activation rules are necessary for the user commands to be analyzed), a server configured to receive a

Art Unit: 2109

request for more than one Web service (page 1, [0007], lines 1-3, multiple Internet services are provided, a service is inherent for the system to receive and process user requests for Internet services), a plurality of watchers, each watcher corresponding to a particular Web service (Abstract, lines 11-14, the watchers are contained within the service automation module), where the service activation rules cause the invocation of at least one of said watchers and corresponding Web services (Abstract, lines 1-8, the Web services are scheduled based on the analysis of the user command, so there are service activation rules and watchers that are able to determine when to invoke a particular Web service), and a scheduler to receive events (page 6, [0039], lines 8-10, the agent scheduler receives a command signal, which is the event).

50. Shim does not disclose that the scheduler compares trigger conditions specified by said service activation rules with trigger information of events, compares state conditions of said service activation rules in the trigger list with a state of the common memory, and selects at least one of said service activation rules, each service activation rule specifying a trigger condition and a state condition for causing a watcher to invoke a particular Web service. However, Auer discloses that the scheduler compares trigger conditions specified by said service activation rules with trigger information of events (Abstract, lines 11-12), compares state conditions of said service activation rules in the trigger list with a state of the common memory (col. 6, lines 38-41, the "pattern portion" corresponds to the state condition of the rule, col. 6, lines 49-51), and selects at least one of said service activation rules (col. 6, lines 51-52), each service activation rule specifying a trigger condition and a state condition (col. 6, lines 49-51).

51. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim and Auer before him or her to incorporate comparisons of trigger and state conditions when deciding which Web services to invoke. The motivation for doing so would have been to allow more granularity in the selection process by using a secondary comparison in the rule matching as taught by Auer (col. 6, lines 49-54). The trigger condition is the primary comparison, and the state condition is the secondary comparison.

52. Neither Shim nor Auer discloses at least one servlet configured to extract a pattern object from the request and to format a response to the request, and a common memory that temporarily stores the pattern object while the Web services specified by the pattern object execute. However, Wu discloses at least one servlet configured to extract a pattern object from the request (page 3, [0042], line 3, "the system parses the form data") and to format a response to the request (page 1, [0016], the system constructs SOAP response messages). Wu further discloses a common memory that temporarily stores the pattern object while the Web services specified by the pattern object execute (page 3, [0052], the system looks up the previous invocation object, which indicates that an object representing a user request is temporarily stored in memory).

53. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer and Wu before him or her to incorporate

Art Unit: 2109

a servlet to extract a pattern object from the request and format a response to the request, and to temporarily store the pattern object while the Web services specified by the pattern object. The motivation for doing so would have been to enable a SOAP message response containing request information to be sent to a JSP page or a log file, and to allow the invocation request to be cloned if a user wants to invoke a previously invoked Web service (Wu, page 3, [0050], page 3, [0052]).

54. As to claim 15, neither Shim nor Auer discloses the system of claim 14, further comprising a termination watcher configured to provide the pattern object back to one of said plurality of servlets to generate a response. However, Wu discloses the system of claim 14, further comprising a termination watcher configured to provide the pattern object back to one of said plurality of servlets to generate a response (page 3, [0046], the queue watcher spurs the client thread, i.e., servlet, after removing the pattern object from the queue, passing the pattern object to the client thread which generates a response, i.e., spurs invocation thread as per configuration, see Fig. 6, #618, #628, and #638).

55. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer and Wu before him or her to provide the pattern object back to one of the servlets to generate a response in order for another thread to execute as taught by Wu (Fig. 6, page 3, [0047]).

56. As to claim 16, neither Shim nor Auer discloses the system of claim 14, wherein said watchers are further configured to modify the pattern object according to

instructions provided from an associated one of the Web services. However, Wu discloses the system of claim 14, wherein said watchers are further configured to modify the pattern object according to instructions provided from an associated one of the Web services (page 2, [0024], the invention dynamically reconfigures Web services by intercepting, transforming, and redirecting SOAP messages, and the SOAP messages are in memory).

57. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer, and Wu before him or her to configure the watchers to modify the pattern object according to instructions from an associated one of the Web services in order to facilitate reporting a status back from the Web service.

58. As to claim 19, Shim discloses a system for processing complex requests for Web services comprising a plurality of service activation rules for causing a watcher to invoke a particular Web service (Abstract, lines 1-8, the Web services are scheduled based on the analysis of the user commands, which implies service activation rules and a watcher that is able to determine when to invoke a particular Web service).

59. Shim does not disclose that each service activation rule specifies a trigger condition and a state condition, nor that said common memory generates events when the content of said common memory is changed, nor does Shim disclose a trigger evaluation processor configured to compare the trigger conditions of the service

Art Unit: 2109

activation rules with trigger information from at least one event. However, Auer discloses that each service activation rule specifies a trigger condition and a state condition (col. 6, lines 49-51), and that said common memory generates events when the content of said common memory is changed (col. 5, lines 66-67, col. 6, lines 1-4, data is sent from the remote system when modifications are made to the remote screen, i.e., remote screen memory).

60. Auer also discloses: a trigger evaluation processor configured to compare the trigger conditions of the service activation rules with trigger information from at least one event (Abstract, lines 11-12, the processor is inherent in the system), that the trigger evaluation processor adds service activation rules that match the at least one event to a trigger list (col. 2, lines 27-30, the processor is inherent in the system), and a state evaluation processor configured to compare the state conditions of service activation rules in the trigger list with a state of the common memory (col. 6, lines 38-41, the "pattern portion" corresponds to the state condition of the rule, col. 6 lines 49-51) and cause the service activation rules of the trigger list that match to be selected (col. 6, lines 51-52, the action portion of the rule is executed).

61. Shim and Auer are analogous art because they are in the same field of endeavor of Web-based systems.

62. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim and Auer before him or her to incorporate the form of the rules used by Auer in the system of Shim, to have the common memory

generates events when the content of said common memory is changed, to have a trigger evaluation processor configured to compare the trigger conditions of the service activation rules with trigger information from at least one event, and to add service activation rules that match event(s) to a trigger list. The motivation for doing so would have been to enable the rules to specify the conditions that the rules-based system would respond to based on the contents of a working memory (Auer, col. 6, lines 49-51), to allow the system to execute the action portion of the rule (Auer, Abstract, lines 11-14), and to enable comparisons to be carried out (Auer, col. 2, lines 27-30 describes the comparisons).

63. Although Auer does not disclose that the state evaluation processor is configured to cause the invocation of at least one watcher and corresponding Web service when a match occurs, Auer discloses that the "action" part of the rule is invoked (Abstract, lines 11-14), and in the combined system, the "action" part of the rules are watchers that invoke a particular Web service.

64. Neither Shim nor Auer disclose a common memory that temporarily stores a pattern object while Web services specified by the pattern object execute. However, Wu discloses a common memory that temporarily stores a pattern object while Web services specified by the pattern object execute (page 3, [0052], the system looks up the previous invocation object, which indicates that an object representing a user request is temporarily stored in memory).

Art Unit: 2109

65. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer and Wu before him or her to temporarily store the pattern object while the Web services specified by the pattern object execute in order to allow the invocation request to be cloned if a user wants to invoke a previously invoked Web service (page 3, [0050], page 3, [0052]).

66. Although Auer does not disclose that a pattern object is what the state evaluation processor matches against, in the combined system of Shim, Auer, and Wu, the data written to memory is stored as a pattern object.

67. As to claim 20, neither Shim, Auer, nor Wu discloses the system of claim 19, wherein said state evaluation processor adds the service activation rules of the trigger list that match the at least one pattern object to an execution list prior to execution of each service activation rule. However, it would have been obvious to one skilled in the art at the time of invention to add the activation rules that match the pattern object to an executable list in order to track the Web services that are being invoked.

68. As to claim 23, Shim discloses a system for scheduling a request for a plurality of Web services comprising:

means for storing a plurality of service activation rules, each service activation rule specifying a trigger condition and a state condition for causing a watcher to invoke a particular Web service (Abstract, lines 1-8, the Web services are scheduled based on the analysis of the user commands, which implies service activation rules and a watcher that is able to determine when to invoke a particular Web service).

Shim does not disclose that the service activation rules specify trigger and state conditions. However, Auer specifies that the service activation rules specify trigger and state conditions (col. 6, lines 49-51).

69. Auer discloses means for receiving at least one event indicating a change in a common memory, wherein each event specifies trigger information (col. 6, lines 49-51), means for comparing the trigger conditions of the service activation rules with the trigger information of the at least one event (Abstract, lines 11-12), means for adding service activation rules that match the at least one event to a trigger list (col. 2, lines 27-30), means for comparing the state conditions of service activation rules in the trigger list with a state of the common memory (col. 6, lines 38-41, the "pattern portion" corresponds to the state condition of the rule, col. 6, lines 49-51), and means for selecting the service activation rules of the trigger list that match the state of the common memory (col. 6, lines 51-52), thereby causing the invocation of at least one watcher and corresponding Web service, wherein the state of the common memory is dictated by at least one pattern object.

70. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim and Auer before him or her to specify trigger and state conditions in the service activation rules, providing means for receiving at least one event indicating a change in a common memory, providing means for adding service activation rules that match the at least one event to a trigger list, providing

means for comparing the state conditions of service activation rules in the trigger list with a state of the common memory, and providing means for selecting the service activation rules of the trigger list that match the state of the common memory. The motivation for doing so would have been to enable the rules to specify the conditions that the rules-based system would respond to, based on the contents of a working memory (Auer, col. 6, lines 49-51), to allow the system to execute the action portion of the rule as taught by Auer (Abstract, lines 11-14), and to allow a second comparison to be used in the rule matching process as taught by Auer (col. 6, lines 49-54).

71. Neither Shim nor Auer disclose that selecting the service activation rules of the trigger list that match the state of the common memory causes the invocation of at least one watcher and corresponding Web service. However, in the combined system of Shim and Auer the invocation of a watcher and corresponding Web service will result from the agent module performing commands based on analyzing the user commands (Shim, page 1, [0008], lines 5-7).

72. Neither Shim nor Auer discloses that the state of the common memory is dictated by at least one pattern object. However, Wu discloses that the state of the common memory is dictated by at least one pattern object (p. 4, [0046], for each user submission concurrency configurations are saved, so data is written to memory in the form of a pattern object, also see p. 1, [0019], first sentence)

73. Shim, Auer, and Wu are analogous arts because they are in the same field of

Art Unit: 2109

endeavor of computer systems.

74. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim and Auer, and the teachings of Wu before him or her to write data to memory in the form of a pattern object, i.e., have at least one pattern object dictate the state of the common memory. The motivation for doing so would have been to keep the data in memory organized which, in turn, makes it easier to work the data.

75. Claims 12, 17, 18, 21, 22, 24, and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shim in view of Auer and Wu, as applied to claims 1, 14, 19, 23, and 26 above, and further in view of Xie et al. (US 2004/0068586), hereinafter Xie.

76. As to claims 12 and 37, Shim discloses the method of claim 1, said method further comprising the steps of:
identifying service activation rules in the execution list corresponding to competitive Web services (page 1, [0008], lines 5-7, the agent module receives and analyzes user command, so the agent module is able to identify the server activation rules), and invoking watchers specified by the identified service activation rules according to said comparing step (page 1, [0008], lines 5-7, the agent module also performs operations according to the commands).

77. Shim does not disclose that said step (f) includes matched service activation rules within an execution list. However, it would have been obvious to one skilled in the art at the time of invention that the service activation rules need to be stored in memory,

Art Unit: 2109

i.e., in a list, for them to be operable in the system.

78. Neither Shim, Auer, nor Wu disclose that the service activation rules correspond to competitive Web services. However, Xie discloses that the service activation rules correspond to competitive Web services (user preference rules are used to map a "semi-generic" request, i.e., a request that specifies the generic type of Web service to a particular Web service, see page 6, [0070], lines 10-13. This implies that there are server activation rules for competing Web services. Otherwise, there would be no need for user preferences to specify a preferred Web service). Xie further discloses comparing the identified service activation rules with at least one service selection rule (user preference rules are used to map a "semi-generic" request, i.e., a request that specifies the generic type of Web service, to invoke a preferred Web service, see page 6, [0070], lines 10-13. The user preference rules are service selection rules that are used to choose the preferred Web service). In addition, the state condition rules of Auer serve as service selection rules (col. 6, lines 38-41, the "pattern portion" corresponds to the state condition of the rule).

79. Shim, Auer, Wu, and Xie are analogous art because they are in the same field of endeavor of computer systems.

80. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer, Wu and Xie before him or her to incorporate service activation rules corresponding to competing Web Services, and to compare the identified service activation rules with at least one service selection rule to invoke a preferred Web service. The motivation for doing so would have been to enable the user to specify user preference rules that are used to choose a preferred Web

service when more than one Web service offering the same utility is available, as taught by Xie (page 6, [0070]).

81. As to claim 17, neither Shim, Auer, nor Wu discloses that the system of claim 14, further comprising an execution evaluation processor configured to select service activation rules associated with competitive Web services according to service selection rules. However, Xie discloses the system of claim 14, further comprising an execution evaluation processor configured to select service activation rules associated with competitive Web services according to service selection rules (page 6, [0070], lines 10-13, the processor, which is inherent, uses user preference rules to map a “semi-generic” request, i.e., a request that specifies the generic type of Web service, to invoke a preferred Web service).

82. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer, Wu, and Xie before him or her to incorporate an execution evaluation processor configured to select service activation rules associated with competitive Web services according to service selection rules. The motivation for doing so would have been to enable the system to “honor” user preferences for particular Web services (Xie, page 6, [0070]).

83. As to claim 18, Shim discloses the system of claim 17, wherein said execution evaluation processor invokes at least one of said watchers according to selected service activation rules. (Abstract, lines 1-8, the processor, which is inherent here,

Art Unit: 2109

invokes Web services based on the analysis of the user commands. The watchers are the tasks called when the service activation rules match a certain user command, and the tasks in turn invoke the Web services. The service activation rules while not expressly described are inherent, because there have to be rules for analyzer to determine which Web services to invoke).

84. As to claim 21, neither Shim, Auer, nor Wu disclose the system of claim 20, further comprising an execution evaluation processor configured to select service activation rules associated with competitive Web services according to service selection rules. However, Xie discloses the system of claim 20, further comprising an execution evaluation processor configured to select service activation rules associated with competitive Web services according to service selection rules (page 6, [0070], lines 10-13, user preference rules are used to map a "semi-generic" request, i.e., a request that specifies the generic type of Web service, to invoke a preferred Web service. The processor is inherent in the system).

85. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer, Wu, and Xie before him or her to incorporate an execution evaluation processor configured to select service activation rules associated with competitive Web services according to service selection rules. The motivation for doing so would have been to enable the system to "honor" user preferences for particular Web services (Xie, page 6, [0070]).

86. As to claim 22, Shim discloses the system of claim 21, wherein said execution

evaluation processor invokes at least one of said watchers according to selected service activation rules (Abstract, lines 1-8, the Web services are scheduled based on the analysis of the user commands, which implies service activation rules and a watcher that invokes a particular Web service when there is a match with the corresponding service activation rule).

87. As to claim 24, Shim discloses the system of claim 23, wherein said means for selecting include matched service activation rules within an execution list, said system further comprising:
means for identifying service activation rules in the execution list corresponding to competitive Web services (page 1, [0008], lines 5-7, the agent module receives and analyzes user commands, so the agent module is able to identify the server activation rules), and means for invoking watchers specified by the identified service activation rules according to said means for comparing the identified service activation rules (Shim does not disclose that the comparison includes at least one service selection rule, however, this will be addressed below. Page 1, [0008], lines 5-7, the agent module also performs operations according the results of the analysis of the commands, implying that tasks associated with the matching rules are invoked and that these tasks invoke the appropriate Web service. The tasks are the watchers).

Although Shim does not explicitly disclose said means for selecting include matched service activation rules within an execution list, it would have been obvious to one skilled in the art at the time of invention to create such an execution list, so that the appropriate task could be called when a match for the service activation rule occurred.

88. Neither Shim, Auer, nor Wu disclose means for comparing the identified service activation rules with at least one service selection rule, and means for invoking watchers specified by the identified service activation rules according to said means for comparing the identified service activation rules. However, Xie discloses means for comparing the identified service activation rules with at least one service selection rule (page 6, [0070], lines 10-13).

89. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Shim, Auer, Wu and Xie before him or her to incorporate the comparison of the identified service activation rules with at least one service selection rule. The motivation for doing so would have been to provide the user with a choice of Web services for a given Web service utility as taught by Xie (page 6, [0070]).

90. Although Shim does not disclose that the comparison done to match a service activation rule includes at least one service selection rule, in the combined system of Shim, Auer, Wu, and Xie, the comparison does include at least one service selection rule.

Conclusion

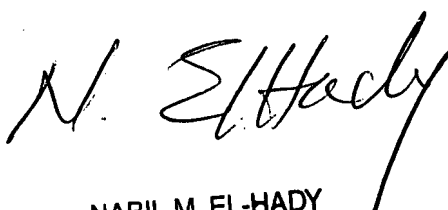
Any inquiry concerning this communication or earlier communications from the examiner should be directed to David A. Holloway whose telephone number is (571)270-1899. The examiner can normally be reached on mon-fri 8:00 am - 5:00 pm (alternate fridays off).

Art Unit: 2109

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nabil El-Hady can be reached on (571)272-3963. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DH 8/9/07


NABIL M. EL-HADY
SUPERVISORY PATENT EXAMINER